

LAPORAN PRAKTIKUM 3

Analisis algoritma



Disusun oleh :

**Difa Bagasputra Maulana
140810180057**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS PADJADJARAN
2020**

Latihan Analisa

1. Untuk $T(n) = 2 + 4 + 6 + 8 + 16 + \dots + n^2$, tentukan nilai C , $f(n)$, n_0 , dan notasi Big-O sedemikian sehingga $T(n) = O(f(n))$ jika $T(n) \leq C$ untuk semua $n \geq n_0$
2. Buktikan bahwa untuk konstanta-konstanta positif p , q , dan r :
 $T(n) = pn^2 + qn + r$ adalah $O(n^2)$, $\Omega(n^2)$, dan $\Theta(n^2)$
3. Tentukan waktu kompleksitas asimptotik (Big-O, Big- Ω , dan Big- Θ) dari kode program berikut:

```
for k ← 1 to n do
  for i ← 1 to n do
    for j ← 1 to n do
       $w_{ij} \leftarrow w_{ij}$  or  $w_{ik}$  and  $w_{kj}$ 
    endfor
  endfor
endfor
```
4. Tulislah algoritma untuk menjumlahkan dua buah matriks yang masing-masing berukuran $n \times n$. Berapa kompleksitas waktunya $T(n)$? dan berapa kompleksitas waktu asimptotiknya yang dinyatakan dalam Big-O, Big- Ω , dan Big- Θ ?
5. Tulislah algoritma untuk menyalin (copy) isi sebuah larik ke larik lain. Ukuran elemen larik adalah n elemen. Berapa kompleksitas waktunya $T(n)$? dan berapa kompleksitas waktu asimptotiknya yang dinyatakan dalam Big-O, Big- Ω , dan Big- Θ ?
6. Diberikan algoritma Bubble Sort sebagai berikut:

```
procedure BubbleSort(input/output  $a_1, a_2, \dots, a_n$ ; integer)
  ( Mengurut tabel integer TabInt[1..n] dengan metode pengurutan bubble-
  sort
  Masukan:  $a_1, a_2, \dots, a_n$ 
  Keluaran:  $a_1, a_2, \dots, a_n$  (terurut menaik)
)
Deklarasi
  k : integer ( indeks untuk traversal tabel )
  pass : integer ( tahapan pengurutan )
  temp : integer ( peubah bantu untuk pertukaran elemen tabel )
Algoritma
  for pass ← 1 to n - 1 do
    for k ← n downto pass + 1 do
      if  $a_k < a_{k-1}$  then
        ( pertukarkan  $a_k$  dengan  $a_{k-1}$  )
        temp ←  $a_k$ 
         $a_k \leftarrow a_{k-1}$ 
         $a_{k-1} \leftarrow temp$ 
      endif
    endfor
  endfor
```

- (a) Hitung berapa jumlah operasi perbandingan elemen-elemen tabel!
- (b) Berapa kali maksimum pertukaran elemen-elemen tabel dilakukan?
- (c) Hitung kompleksitas waktu asimptotik (Big-O, Big- Ω , dan Big- Θ) dari algoritma Bubble Sort tersebut!

7. Untuk menyelesaikan problem X dengan ukuran N tersedia 3 macam algoritma:

- (a) Algoritma A mempunyai kompleksitas waktu $O(\log N)$
- (b) Algoritma B mempunyai kompleksitas waktu $O(N \log N)$
- (c) Algoritma C mempunyai kompleksitas waktu $O(N^2)$

Untuk problem X dengan ukuran $N=8$, algoritma manakah yang paling cepat? Secara asimptotik, algoritma manakah yang paling cepat?

8. Algoritma mengevaluasi polinom yang lebih baik dapat dibuat dengan metode Horner berikut:

$$p(x) = a_0 + x(a_1 + x(a_2 + x(a_3 + \dots + x(a_{n-1} + a_n x))))$$

```
function p2(input x : real) → real  
( Mengembalikan nilai  $p(x)$  dengan metode Horner)
```

Deklarasi

```
k : integer  
b1, b2, ..., bn : real
```

Algoritma

```
bn ← an  
for k ← n - 1 downto 0 do  
    bk ← ak + bk+1 * x  
endfor  
return b0
```

Hitunglah berapa operasi perkalian dan penjumlahan yang dilakukan oleh algoritma diatas, Jumlahkan kedua hitungan tersebut, lalu tentukan kompleksitas waktu asimptotik (Big-O)nya. Manakah yang terbaik, algoritma p atau p2?

Jawaban

$$\begin{aligned} \textcircled{1} \quad T(n) &= 2 + 4 + 8 + 16 + \dots + 2^n \\ &= \frac{2(2^n - 1)}{2 - 1} = 2(2^n - 1) = 2^{n+1} - 2 \end{aligned}$$

$$T(n) = 2^{n+1} - 2 = O(2^n)$$

$$T(n) \leq C \cdot f(n)$$

$$2^{n+1} - 2 \leq C \cdot 2^n$$

$$2 \cdot 2^n - 2 \leq C \cdot 2^n$$

$$2 - \frac{2}{2^n} \leq C$$

$$\begin{aligned} n_0 &= 1 \\ 2 - \frac{2}{2} &\leq C \end{aligned}$$

$$\boxed{C \geq 1}$$

$$\boxed{\begin{aligned} f(n) &= 2^n \\ n_0 &= 1 \end{aligned}}$$

$$\textcircled{2} \quad T(n) = Pn^2 + qn + r$$

$$\rightarrow \Theta(n^2) \rightarrow \text{Big } O$$

$$T(n) \leq C \cdot f(n)$$

$$Pn^2 + qn + r \leq C \cdot n^2 \quad \left\{ \begin{aligned} n_0 &= 1 \\ P + q + r &\leq C \end{aligned} \right.$$

$$P + \frac{q}{n} + \frac{r}{n^2} \leq C \quad \left\{ \begin{aligned} C &\geq P + q + r \end{aligned} \right.$$

$$\rightarrow \Omega(n^2) \rightarrow \text{Big } \Omega$$

$$T(n) \geq C \cdot g(n)$$

$$Pn^2 + qn + r \geq C \cdot n$$

$$Pn + q + \frac{r}{n} \geq C$$

$$\left\{ \begin{aligned} n_0 &= 1 \\ P + q + r &\geq C \\ C &\leq P + q + r \end{aligned} \right.$$

$$\rightarrow \text{Big } \Theta(n^2)$$

karena keduanya berderajat sama, maka $\Theta(n^2)$ terbukti benar

$$\textcircled{3} \quad \text{for } k \leftarrow 1 \text{ to } n \text{ do}$$

$$\quad \text{for } i \leftarrow 1 \text{ to } n \text{ do}$$

$$\quad \quad \text{for } j \leftarrow 1 \text{ to } n \text{ do}$$

$$\quad \quad \quad w_{ij} \leftarrow w_{ij} \text{ or } w_{ik} \text{ and } w_{kj} \rightarrow$$

$$\quad \quad \text{end for}$$

$$\quad \text{end for}$$

$$\text{end for}$$

$$\boxed{\begin{aligned} n \cdot n \cdot n \\ T(n) &= n^3 \end{aligned}}$$

* Big O

$$n^3 \leq c \cdot n^3$$

$$\rightarrow n_0 = 1$$

$$1 \leq c \cdot 1$$

$$c \geq 1$$

* Big Ω

$$n^3 \geq c \cdot n^3$$

$$\rightarrow n_0 = 1$$

$$1 \geq c \cdot 1$$

$$c \leq 1$$

* Big Θ

karena $O(n^3)$ dan $\Omega(n^3)$
bersifat sama maka $\Theta(n^3)$
terbukti benar

5. Algoritma menyalin larik

for $i \leftarrow 1$ to n do

$a_i \leftarrow b_i \rightarrow n \rightarrow T(n) = n$

end for

* Big O

$$n \leq c \cdot n$$

$$\rightarrow n_0 = 1$$

$$1 \leq c \cdot 1$$

$$c \geq 1$$

* Big Ω

$$n \geq c \cdot n$$

$$\rightarrow n_0 \geq 1$$

$$1 \geq c \cdot 1$$

$$c \leq 1$$

* Big Θ

karena $O(n)$ dan $\Omega(n)$
bersifat sama, maka $\Theta(n)$
terbukti benar

6. (a) Jumlah operasi perbandingan

$$1 + 2 + 3 + \dots + (n-1)$$

$$= \frac{n(n-1)}{2} \text{ kali}$$

(b) Berapa kali maksimum pertukaran elemen tabel dilakukan

$$\frac{n(n-1)}{2} \text{ kali}$$

(c) Kompleksitas

* Best Case (semua data sudah terurut)

$$\frac{(n-1)n}{2} \text{ kali}, T_{\min}(n) = \frac{n(n-1)}{2} = \frac{n^2 - n}{2}$$

* Worst Case (semua data terurut terbalik)

$$\text{perbandingan} \rightarrow \frac{n(n-1)}{2}$$

$$\text{Assignment} \rightarrow \frac{3n(n-1)}{2}$$

$$T_{\max}(n) = \frac{4n(n-1)}{2} = 2n^2 - 2n$$

→ Big O

$$2n^2 - 2n \leq c(n^2)$$

$$2 - \frac{2}{n} \leq c$$

$$n_0 = 1 \rightarrow 2 - 2 \leq c$$

$$c \geq 0$$

→ Big Ω

$$\frac{n^2 - n}{2} \geq c(n^2)$$

$$\frac{1}{2} - \frac{1}{2n} \geq c$$

$$n_0 = 1 \rightarrow \frac{1}{2} - \frac{1}{2} \geq c$$

$$c \leq 0$$

7. a. Algoritma A $O(\log N)$
 b. Algoritma B $\rightarrow O(N \log N)$
 c. Algoritma C $\rightarrow O(N^2)$

Misal $N = 8$

maka algoritma

$$a. A \rightarrow O(\log 8) = O(3 \log 2)$$

$$b. \text{Algoritma B} \rightarrow O(8 \log 8) = O(24 \log 2)$$

$$c. \text{Algoritma C} \rightarrow O(8^2) = O(64)$$

yang paling efektif adalah algoritma A karena semakin kecil nilai dalam $O()$ semakin baik operasi yang digunakan

8. Operasi Assignment

- $b_n \leftarrow a_n = 1$ kali
- $b_n \leftarrow a_n + b_{n+1} \times X = n$ kali

$$T(n) = n + 1$$

$$O(n) = \text{untuk } p^2$$

Algoritma P

$$\text{penjumlahan} = \underline{n \text{ kali}}$$

$$\text{perkalian} = \underline{n \text{ kali}}$$

$$T(n) = 2n$$

Algoritma p^2 lebih baik daripada P